

CLAIMS

1. A method for releasing a product under development, the method comprising:

5 storing in memory an association between a version (hereinafter "current version") of a component of the product and a time (hereinafter "future tick") selected from periodically recurring times in future; and

10 identifying a version (hereinafter "release version") of the component as being available for release, using the association and a time (hereinafter "most recent tick") that occurred most recently in past, from among periodically recurring times.

2. The method of Claim 1 wherein:

15 the release version is different from the current version if the future tick is yet to occur.

3. The method of Claim 1 further comprising:

20 the release version is the current version if the future tick is the most recent tick.

4. The method of Claim 1 further comprising:

25 on passage of said future tick (hereinafter "past tick"), storing in memory another association, of said version with a time in future (hereinafter "next tick") that is scheduled to occur immediately after said past tick, among the periodically recurring times.

30 5. The method of Claim 1 wherein the association is stored in a record of a database.

6. The method of Claim 1 further comprising:

storing an identity of a person responsible for  
development of the current version.

7. The method of Claim 1 wherein the component comprises  
5 software, the method further comprising:

storing an address of the current version.

8. The method of Claim 1 wherein the component comprises  
software, the method further comprising:

10 copying the software to a central location of storage  
of other components.

9. The method of Claim 1 further comprising:

15 storing an identity of a bug that has been fixed in the  
current version.

10. The method of Claim 1 further comprising:

20 storing a label of the component used in a version  
control system.

11. The method of Claim 1 further comprising:

storing an indicator (also called "staging time") of  
when the current version is associated with the future tick.

25 12. The method of Claim 1 further comprising:

storing for the current version an identity of a  
release of all components in which the current version is to  
be included.

30 13. The method of Claim 12 wherein:

the release is predetermined to occur subsequent to a  
time (hereinafter "milestone tick") selected from the  
periodically recurring times.

14. The method of Claim 13 wherein:

said future tick is one of a plurality of ticks prior to the milestone tick.

5

15. The method of Claim 1 further comprising:

receiving said future tick and an identification of said version via a graphical user interface.

10 16. The method of Claim 1 further comprising:

storing an association of a unique identifier with the current version.

17. The method of Claim 16 further comprising:

15 receiving the unique identifier from a person responsible for development of the version.

18. The method of Claim 1 wherein:

said periodically recurring times occur once a week.

20

19. The method of Claim 18 wherein:

each tick occurs on a predetermined day selected from a group consisting of Tuesday, Wednesday and Thursday.

25 20. The method of Claim 18 wherein:

each tick occurs on Wednesday.

21. The method of Claim 1 wherein said current version is hereinafter "first version", the method further comprising:

30 storing in memory another association between another version (hereinafter "second version") of the component and said future tick;

wherein after the future tick has occurred, said second version is identified as the release version.

22. The method of Claim 21 further comprising:

- 5       storing a first association of a first identifier with the first version; and  
          storing a second association of a second identifier with the second version.

10   23. The method of Claim 21 wherein:

          said storing of another association is performed prior to said future tick.

24. The method of Claim 21 wherein:

- 15       said storing of another association is performed subsequent to said future tick only as an exception.

25. The method of Claim 1 wherein said current version is hereinafter "first version," said association is hereinafter "first association," and said "future tick" is hereinafter "first tick," the method further comprising:

- 20       storing in memory a second association between another version (hereinafter "second version") of the component and a second tick that is subsequent to said first tick;  
25       wherein said first tick has already occurred and said second tick is yet to occur.

26. A computer readable storage medium encoded with software instructions to perform the method of Claim 1 when executed  
30   by a computer.

27. A signal embedded in a carrier medium and encoded with software instructions to perform the method of Claim 1 when executed by a computer.

5 28. A computer programmed to track each component of a product under development, the computer comprising:

memory holding a version of a component and a time (hereinafter "tick") that is one of several periodically recurring times in future, at memory locations addressed by  
10 a data structure;

first means for determining from current time, a time (hereinafter "most recent tick") that occurred most recently in past, from among periodically recurring times; and

second means coupled to the first means to receive  
15 therefrom the most recent tick, and coupled to the memory to receive therefrom one or more pairs (of version and tick) addressed by the data structure, the second means identifying a version (hereinafter "release version") as being available for release if the tick paired to the  
20 release version matches the most recent tick.

29. The computer programmed as in Claim 28 wherein:

said memory also holds an identity of a person responsible for development of said version, an address of  
25 the version, an identity of a bug that has been fixed in the version, a label of the component used in a version control system for the version, an indicator of when the version is associated with the tick, and an identity of a release of all components in which the version is to be included.

30